



# Optimization problems in multiple subtree graphs<sup>☆</sup>

Danny Hermelin<sup>a,\*</sup>, Dror Rawitz<sup>b,\*</sup>

<sup>a</sup> Department of Computer Science, University of Haifa, Haifa 31905, Israel

<sup>b</sup> School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel

## ARTICLE INFO

### Article history:

Received 1 February 2009

Received in revised form 28 February 2010

Accepted 19 March 2010

Available online 13 April 2010

### Keywords:

Approximation algorithms

Minimum dominating set

Maximum independent set

Minimum coloring

Minimum vertex cover

Maximum clique

Multiple subtree graphs

## ABSTRACT

We study various optimization problems in *t*-subtree graphs, the intersection graphs of *t*-subtrees, where a *t*-subtree is the union of *t* disjoint subtrees of some tree. This graph class generalizes both the class of chordal graphs and the class of *t*-interval graphs, a generalization of interval graphs that has recently been studied from a combinatorial optimization point of view. We present approximation algorithms for the MAXIMUM INDEPENDENT SET, MINIMUM COLORING, MINIMUM VERTEX COVER, MINIMUM DOMINATING SET, and MAXIMUM CLIQUE problems.

© 2010 Elsevier B.V. All rights reserved.

## 1. Introduction

Geometric intersection graphs are a very popular topic in algorithmic graph theory. This is because of the many natural applications they model, and due to the rich combinatorial structure that comes along with most of them, allowing for numerous algorithmic techniques and frameworks. Two of the oldest and most well-studied graph classes in this area are the class of interval graphs, intersection graphs of intervals of a line, and the class of chordal graphs, intersection graphs of subtrees of a tree [10]. Many classical NP-complete problems become polynomial-time solvable in both these classes of graphs (for more details see [11] and references therein).

In [3,6], various optimization problems were considered in the class of *t*-interval graphs, a natural generalization of interval graphs. These are defined as intersection graphs of *t*-intervals, which are 1-dimensional objects formed by taking the union of *t* disjoint intervals. In this paper, we study the class of *t*-subtree graphs, which generalizes the class of *t*-interval graphs by replacing intervals with subtrees. Thus, *t*-subtree graphs form a hybrid between *t*-interval graphs and chordal graphs, and provide a natural generalization for these two graph classes.

We consider various classical optimization problems in *t*-subtree graphs. Given a *t*-subtree graph *G* along with its *t*-subtree representation  $\mathcal{S}$  (a formal definition is given in Section 2), we present approximation algorithms for the following problems:

- **MINIMUM DOMINATING SET:** Find a minimum weight subset  $\mathcal{S}' \subseteq \mathcal{S}$  such that for each *t*-subtree  $S \in \mathcal{S}$  there is a *t*-subtree  $S' \in \mathcal{S}'$  which intersects *S*.

<sup>☆</sup> This work was done at the Caesarea Edmond Benjamin de Rothschild Foundation Institute (CRI), University of Haifa. An extended abstract was presented at the 7th Workshop on Approximation and Online Algorithms (WAOA), 2009.

\* Corresponding author. Fax: +972 3 6407095.

E-mail addresses: [danny@cri.haifa.ac.il](mailto:danny@cri.haifa.ac.il) (D. Hermelin), [rawitz@eng.tau.ac.il](mailto:rawitz@eng.tau.ac.il), [rawitz@cri.haifa.ac.il](mailto:rawitz@cri.haifa.ac.il) (D. Rawitz).

- **MAXIMUM INDEPENDENT SET:** Find a maximum weight pairwise non-intersecting subset  $\mathcal{S}' \subseteq \mathcal{S}$ .
- **MINIMUM COLORING:** Partition  $\mathcal{S}$  into the smallest number of subsets such that each subset is pairwise non-intersecting.
- **MINIMUM VERTEX COVER:** Find a minimum weight subset  $\mathcal{S}' \subseteq \mathcal{S}$  such that  $\mathcal{S} \setminus \mathcal{S}'$  is pairwise non-intersecting.
- **MAXIMUM CLIQUE:** Find a maximum weight pairwise intersecting subset  $\mathcal{S}' \subseteq \mathcal{S}$ .

### Related work

Gavril [9] showed that the above optimization problems, with the exception of MINIMUM DOMINATING SET, can be solved in polynomial-time in chordal graphs. On the other hand, MINIMUM DOMINATING SET in chordal graphs was shown to be NP-hard in [5]. Recently, Chlebík and Chlebíková showed that MINIMUM DOMINATING SET in chordal graphs cannot be approximated to within a factor of  $(1 - \varepsilon) \ln n$  in polynomial time, for any constant  $\varepsilon > 0$ , unless  $\text{NP} \subseteq \text{DTIME}(n^{O(\log \log n)})$ .

In  $t$ -interval graphs all problems are computationally hard. Griggs and West [12] showed that the class of graphs with maximum degree  $\Delta$  are  $\lceil (\Delta + 1)/2 \rceil$ -interval graphs. It follows that MINIMUM DOMINATING SET and MINIMUM VERTEX COVER are APX-hard, for  $t \geq 2$  [14], and that MINIMUM COLORING is NP-hard, for  $t \geq 3$  [8]. MAXIMUM CLIQUE was shown to be NP-hard for  $t \geq 3$  in [6]. Bar-Yehuda et al. [3] showed that, for  $t \geq 2$ , MAXIMUM INDEPENDENT SET in  $t$ -interval graphs is APX-hard, and that it cannot be approximated within a factor of  $O(t / \log t)$  unless  $P = \text{NP}$ . Finally, it is NP-hard to determine whether a given graph is  $t$ -interval for  $t \geq 2$  [16].

### Our results

In this paper we present approximation algorithms for optimization problems in  $t$ -subtree graphs. For MAXIMUM INDEPENDENT SET, MINIMUM COLORING, MINIMUM VERTEX COVER, and MAXIMUM CLIQUE we obtain approximation ratios of  $2t$ ,  $2t$ ,  $2 - 1/t$ , and  $(t^2 - t + 1)/2$ , respectively, which match the approximation ratios of the corresponding algorithms for  $t$ -interval graphs in [3,6] by extending these in a natural manner. We note that similarly to the algorithm for MINIMUM VERTEX COVER in  $t$ -interval graphs from [6], our algorithm for MINIMUM VERTEX COVER works even without a  $t$ -subtree representation of the input graph. As mentioned above, MINIMUM DOMINATING SET is different in that it is hard to approximate even in chordal graphs (i.e. 1-subtree graphs). Therefore we consider the case where each multiple subtree has at most  $\ell$  leaves, and provide an  $\ell^2$ -approximation algorithm for this case. We also show that MINIMUM DOMINATING SET in multiple subtree graphs is NP-hard to approximate within  $\ell - 1 - \varepsilon$ , for any  $\varepsilon > 0$ .

## 2. Preliminaries

All graphs in this paper are simple and undirected. As usual, we denote the vertex-set and edge-set of a given graph  $G$  by  $V(G)$  and  $E(G)$ , respectively. For a graph  $G$  and a subset of vertices  $V \subseteq V(G)$ , we let  $G - V$  denote the graph obtained by deleting all vertices of  $V$  from  $G$ , and by  $G[V]$  the subgraph that consists of the vertices in  $V$  and the edges connecting them, namely  $G[V] = G - (V(G) \setminus V)$ . For a vertex  $v \in V(G)$ , we let  $N(v)$  denote the set of neighbors of  $v$ , i.e.  $N(v) = \{u \in V : (v, u) \in E\}$ , and we let  $N[v] = N(v) \cup \{v\}$ .

Let  $\mathcal{T}$  be an infinite rooted tree with vertices denoted by Greek letters. A set of vertices  $T \subseteq V(\mathcal{T})$  is a *subtree* of  $\mathcal{T}$  if  $\mathcal{T}[T]$  is a tree. Two subtrees  $T$  and  $T'$  *intersect*, denoted  $T \cap T' \neq \emptyset$ , if they share a common vertex, and otherwise they are *disjoint*. A set  $S$  of at most  $t$  subtrees of  $\mathcal{T}$  is called a  $t$ -subtree over  $\mathcal{T}$ . Two  $t$ -subtrees  $S$  and  $S'$  *intersect*, denoted  $S \cap S' \neq \emptyset$ , if there is a subtree  $T \in S$  which intersects a subtree  $T' \in S'$ , and they are *disjoint* whenever they are non-intersecting. If  $\alpha \in T$  for a subtree  $T$  in a  $t$ -subtree  $S$ , we slightly abuse notation by writing  $\alpha \in S$ .

A family of  $t$ -subtrees  $\mathcal{S}$  is a *representation* of some graph  $G$  if there exists a bijective correspondence  $v \rightarrow S_v$  from the vertices of  $G$  to the  $t$ -subtrees in  $\mathcal{S}$  such that  $(u, v) \in E(G)$  if and only if  $S_u \cap S_v \neq \emptyset$ . In this case,  $G$  is the *intersection graph* of  $\mathcal{S}$ , and thus a  $t$ -subtree graph, and we write this as  $G = G_{\mathcal{S}}$ . We will be considering weighted  $t$ -subtree graphs, i.e.  $t$ -subtree graphs  $G$  with weight functions  $w : V(G) \rightarrow \mathbb{Q}^+$ .

Let  $\Pi$  be an optimization problem, and let  $A$  be an algorithm for  $\Pi$ . Denote by  $A(I)$  the value of the solution computed by  $A$  for the instance  $I$  of  $\Pi$ , and let  $\text{OPT}(I)$  denote the optimal value for the instance  $I$ . The solution computed by  $A$  for an instance  $I$  is said to be  $r$ -approximate, if  $A(I) \leq \text{OPT}(I) \cdot r$ , if  $\Pi$  is a minimization problem, and  $A(I) \geq \text{OPT}(I)/r$ , if  $\Pi$  is a maximization problem.  $A$  is an  $r$ -approximation algorithm for  $\Pi$  if it computes an  $r$ -approximate solution for any instance  $I$  of  $\Pi$ . In this case we say that the *approximation ratio* of algorithm  $A$  is  $r$ . For example, an  $r$ -approximation algorithm for MAXIMUM INDEPENDENT SET returns a pairwise non-intersecting subset of  $t$ -subtrees which has total weight at least  $r$  times the optimal, for any  $t$ -subtree graph.

## 3. Minimum dominating set

We begin with MINIMUM DOMINATING SET. We show that MINIMUM DOMINATING SET in  $t$ -subtree graphs is NP-hard to approximate within  $\ell(\mathcal{S}) - 1 - \varepsilon$ , for any  $\varepsilon > 0$ , where  $\ell(\mathcal{S})$  is the maximum total number of leaves in any  $t$ -subtree belonging to  $\mathcal{S}$ . On the positive side, we present an  $\ell(\mathcal{S})^2$ -approximation algorithm that extends the  $t^2$ -approximation algorithm for MINIMUM DOMINATING SET in  $t$ -interval graphs [6]. This algorithm is based on a reduction to the MINIMUM PATH HITTING problem, and on the approximation algorithm of Parekh and Segev [15] given for this problem.

### 3.1. Approximation lower bounds

We present a reduction from the special case of MINIMUM SET COVER in which each element appears in at most  $t$  sets, i.e. the  $t$ -MINIMUM SET COVER problem, to MINIMUM DOMINATING SET in  $t$ -subtree graphs. We note that all subtrees in the resulting  $t$ -subtree instance are paths.

**Lemma 1.** *For every positive integer  $t \in \mathbb{N}$ , there is an approximation preserving reduction from  $t$ -MINIMUM SET COVER to MINIMUM DOMINATING SET in  $t$ -subtree graphs.*

**Proof.** Let  $(X, \mathcal{C})$  be a MINIMUM SET COVER instance, where  $X = \{x_1, \dots, x_n\}$  is a universe of  $n$  elements, and  $\mathcal{C} = \{C_1, \dots, C_m\}$  is the family of  $m$  subsets of  $X$ . We assume without loss of generality that  $\bigcup_j C_j = X$ . We construct a tree  $\mathcal{T}$  and a family  $\mathcal{S}$  of subtrees as follows. The tree  $\mathcal{T}$  consists of a root  $\alpha_0$  adjacent to  $m$  vertex-disjoint paths  $\alpha_{j1}, \dots, \alpha_{jn}$ , for  $j \in \{1, \dots, m\}$ . For each subset  $C_j \in \mathcal{C}$ , we designate the subtree  $T_j = \{\alpha_0, \alpha_{j1}, \dots, \alpha_{jn}\}$ . We note that the node  $\alpha_{ji}$  is reserved for  $x_i$ , but will be used by the subtree that corresponds to  $x_i$  only if  $x_i \in C_j$ . Now the set of  $t$ -subtrees  $\mathcal{S}$  consists of  $n + m$   $t$ -subtrees, where  $S_i = \{\{\alpha_{ji} : x_i \in C_j\}\}$ , for  $i \in \{1, \dots, n\}$ , and  $S_{n+j} = \{T_j\}$ , for  $j \in \{1, \dots, m\}$ . Observe that  $\mathcal{S}$  is a  $t$ -subtree family, and that it can be constructed in polynomial-time. We argue that a dominating set in  $G = G_{\mathcal{S}}$ , the intersection graph of  $\mathcal{S}$ , corresponds to a set cover of  $X$  of equal size, and vice-versa.

Let  $\mathcal{C}' \subseteq \mathcal{C}$  be a set cover of  $X$ . We claim that  $D = \{v_{n+j} : C_j \in \mathcal{C}'\}$  is a dominating set in  $G$ . First, observe that all vertices  $v_{n+j}$ , where  $1 \leq j \leq m$ , dominate each other since each  $S_{n+j}$  includes the root  $\alpha_0$ . Also, if a vertex  $v_i$  is not dominated by  $D$ , then  $x_i$  is not covered by any set in  $\mathcal{C}'$ . Conversely, let  $D$  be a dominating set in  $G$ . Without loss of generality, we may assume that  $v_i \notin D$  for all  $1 \leq i \leq n$ , since if  $x_i \in C_j$  then we can replace  $v_i$  with  $v_{n+j}$  in  $D$ . Thus, since all  $t$ -subtrees that correspond to elements are dominated, the set  $\mathcal{C}' = \{C_j : S_{n+j} \in D\}$  is a set cover of  $X$ .  $\square$

It is known that it is NP-hard to approximate  $t$ -MINIMUM SET COVER within  $t - 1 - \varepsilon$  for any  $\varepsilon > 0$  [7]. Since  $\ell(\mathcal{S}) = t$  in the above construction, we obtain the following:

**Corollary 1.** *MINIMUM DOMINATING SET in  $t$ -subtree graphs is NP-hard to approximate within  $t - 1 - \varepsilon$ , for any  $\varepsilon > 0$ , even in the special case where each  $t$ -subtree consists of either a single path or  $t$  vertices. Furthermore, it is NP-hard to approximate within  $\ell(\mathcal{S}) - 1 - \varepsilon$ , for any  $\varepsilon > 0$ .*

### 3.2. Bounded number of leaves

We next present an  $\ell(\mathcal{S})^2$ -approximation algorithm for MINIMUM DOMINATING SET in multiple subtree graphs. As a first step, we show that we can consider edge intersections in the subtrees, rather than vertex intersections.

**Lemma 2.** *Let  $G$  be a  $t$ -subtree graph with a  $t$ -subtree representation  $\mathcal{S}$  over a tree  $\mathcal{T}$ . Then  $G$  has a  $t$ -subtree representation  $\mathcal{S}^*$  over another tree  $\mathcal{T}^*$ , with  $(u, v) \in E(G)$  if and only if the  $t$ -subtrees corresponding to  $u$  and  $v$  in  $\mathcal{S}^*$  share an edge. Furthermore,  $\mathcal{S}^*$  can be computed in polynomial-time, and  $\ell(\mathcal{S}^*) = \ell(\mathcal{S})$ .*

**Proof.** We construct a  $t$ -subtree representation  $\mathcal{S}^*$  over  $\mathcal{T}^*$  as follows. First,  $\mathcal{T}^*$  is obtained by splitting every vertex  $\alpha$  of  $\mathcal{T}$  into two adjacent vertices:

$$\begin{aligned} V(\mathcal{T}^*) &= \{\alpha' : \alpha \in V(\mathcal{T})\} \cup \{\alpha'' : \alpha \in V(\mathcal{T})\} \\ E(\mathcal{T}^*) &= \{(\alpha'', \beta') : (\alpha, \beta) \in E(\mathcal{T})\} \cup \{(\alpha', \alpha'') : \alpha \in V(\mathcal{T})\}. \end{aligned}$$

Second, for every subtree  $T$  belonging to some  $t$ -subtree in  $\mathcal{S}$ , we define the associated subtree  $T^*$  with  $V(T^*) = \{\alpha' : \alpha \in V(T)\} \cup \{\alpha'' : \alpha \in V(T)\}$ . Each  $t$ -subtree  $S = \{T_1, \dots, T_t\}$  in  $\mathcal{S}$  is then associated with the  $t$ -subtree  $S^* = \{T_1^*, \dots, T_t^*\}$ . It is not hard to verify that two subtrees  $S_1$  and  $S_2$  contain a node  $\alpha$  if and only if the subtrees induced by  $S_1^*$  and  $S_2^*$  contain the edge  $(\alpha', \alpha'')$ . Clearly,  $\mathcal{S}^*$  can be computed in polynomial-time, and in addition  $\ell(\mathcal{S}^*) = \ell(\mathcal{S})$  by construction.  $\square$

Next, we define a more general variant of MINIMUM DOMINATING SET in  $t$ -subtree graphs. In this variant, we are given two (not necessarily disjoint) families of  $t$ -subtrees, a red family  $\mathcal{R} = \{R_1, \dots, R_n\}$  and a blue family  $\mathcal{B} = \{B_1, \dots, B_m\}$ , and our goal is to find a minimum weight subset  $\mathcal{R}' \subseteq \mathcal{R}$  which dominates  $\mathcal{B}$ , i.e. every  $t$ -subtree  $B \in \mathcal{B}$  is intersected by some  $R \in \mathcal{R}'$ . (We assume that  $\mathcal{R}$  dominates  $\mathcal{B}$ .) Notice that when  $\mathcal{B} = \mathcal{R}$ , we return to MINIMUM DOMINATING SET in  $t$ -subtree graphs.

The extended variant of MINIMUM DOMINATING SET in  $t$ -subtree graphs can be formulated using the following linear integer program:

$$\begin{aligned} \min \quad & \sum_{R \in \mathcal{R}} w(R)x(R) \\ \text{s.t.} \quad & \sum_{R: B \cap R \neq \emptyset} x(R) \geq 1 \quad \forall B \in \mathcal{B} \\ & x(R) \in \{0, 1\} \quad \forall R \in \mathcal{R} \end{aligned} \tag{DS}$$

where  $x(R)$  is a variable corresponding  $R \in \mathcal{R}$  and is interpreted as  $x(R) = 1$  if and only if  $R$  is taken to the solution. The constraint that corresponds to a blue  $t$ -subtree  $B$  guarantees that at least one red subtree that shares an edge with  $B$  is contained in the solution. We use  $\sqcap$  instead of  $\cap$  to remind the reader that we consider edge intersections, rather than vertex intersections. The linear programming relaxation of DS is obtained by replacing the integrality constraints by:  $x(R) \geq 0$ , for every  $R \in \mathcal{R}$ .

We will need the notion of descending paths and  $\ell$ -paths in  $\mathcal{T}$ . Recall that  $\mathcal{T}$  is rooted. A *descending path* in  $\mathcal{T}$  is a path  $\alpha_1, \alpha_2, \dots, \alpha_p$ , where  $\alpha_i$  is an ancestor of  $\alpha_j$  for  $i < j$ . An  $\ell$ -path  $P$  in  $\mathcal{T}$  is a collection of at most  $\ell$  descending paths which are pairwise edge-disjoint. An  $\ell$ -path representation of a  $t$ -subtree graph  $G$ , is a representation which consists only of  $\ell$ -paths.

The next lemma can be proved by induction on the number of leaves:

**Lemma 3.** *If  $G$  is a  $t$ -subtree graph with a  $t$ -subtree representation  $\delta$ , then there exists an  $\ell$ -path representation of  $G$  with  $\ell = \ell(\delta)$ .*

**Proof.** Let  $S$  be a  $t$ -subtree with  $\ell(S)$  leaves. We prove that  $S$  can be represented by at most  $\ell(S)$  descending paths using induction on the number of leaves. In the base case,  $S$  has only one leaf, which means that  $S$  contains one descending path. For the inductive step, pick a leaf  $\alpha_0$  in a subtree  $T \in S$ . Let  $\alpha_0, \dots, \alpha_k$  be a maximum length path such that  $\alpha_{i-1}$  is an only child of  $\alpha_i$ , for every  $i \in \{1, \dots, k\}$ , and let  $S'$  be the  $t$ -subtree obtained by removing  $\alpha_0, \dots, \alpha_k$  from  $S$ . Since  $S'$  has  $\ell(S) - 1$  leaves, it has an  $(\ell(S) - 1)$ -path representation by the inductive hypothesis. By adding  $\alpha_0, \dots, \alpha_k$  we obtain an  $\ell(S)$ -path representation of  $S$ .  $\square$

In the rest of this section we design an approximation algorithm for (DS) in graphs with  $\ell$ -path representations (where intersections are by edges). We first address the case where both  $\mathcal{R}$  and  $\mathcal{B}$  contain 1-paths. In [15], Parekh and Segev devised a 4-approximation algorithm for the MINIMUM PATH HITTING problem using a reduction to this special case of (DS). Formally, they obtained the following result:

**Lemma 4 ([15]).** *If  $\mathcal{R}$  and  $\mathcal{B}$  consist of 1-subtrees, each of which contains a single descending path, then the LP-relaxation of (DS) has an integral optimal solution, and this solution can be computed in polynomial time.*

We next show how to use this result to obtain an approximation algorithm in the case of  $\ell > 1$ . Let  $\mathcal{R}$  and  $\mathcal{B}$  be two families of  $\ell$ -paths. We may assume without loss of generality that all  $\ell$ -paths in  $\mathcal{R}$  and  $\mathcal{B}$  contain exactly  $\ell$  paths. Now, let  $x^*$  denote the corresponding optimal solution of the LP-relaxation of DS.

For a descending path  $P$  in a blue  $\ell$ -path  $B \in \mathcal{B}$ , let  $\mathcal{R}(P) \subseteq \mathcal{R}$  denote that subset of red  $\ell$ -paths that have descending paths intersecting  $P$ . For each  $B \in \mathcal{B}$ , we select a unique *representative* descending path  $P_B \in B$  such that  $P_B = \operatorname{argmax}_{P \in B} \sum_{R \in \mathcal{R}(P)} x^*(R)$ . In other words, the representative  $P_B$  is the maximum dominated descending path of  $B$ . For a red  $\ell$ -path  $R \in \mathcal{R}$ , we let  $P_R^i$  denote the  $i$ th path in  $R$ , for  $1 \leq i \leq \ell$ . We construct a new 1-path instance  $(\mathcal{R}', \mathcal{B}', w')$  by taking the representatives of the blue  $\ell$ -path to be  $\mathcal{B}'$ , i.e.  $\mathcal{B}' = \{P_B : B \in \mathcal{B}\}$ , and defining  $\mathcal{R}' = \{P_R^i : R \in \mathcal{R}, i = 1, \dots, \ell\}$  with  $w'(P_R^i) = w(R)/\ell$  for  $P_R^i \in \mathcal{R}'$ . Next we use Lemma 4 to obtain an integral optimal solution  $x'$  of the LP-relaxation of (DS) with respect to the new instance  $(\mathcal{R}', \mathcal{B}', w')$ . We output the solution  $\mathcal{D} \subseteq \mathcal{R}$  defined by  $\mathcal{D} = \{R : x'(P_R^i) = 1 \text{ for some } i\}$ .

**Lemma 5.**  *$\mathcal{D}$  is an  $\ell^2$ -approximate dominating set of  $\mathcal{B}$ .*

**Proof.** First observe that  $\mathcal{D}$  dominates  $\mathcal{B}$ , since all representatives are dominated by  $\mathcal{D}$ . Let  $x$  be the incidence vector of  $\mathcal{D}$ . Note that

$$\sum_{R \in \mathcal{R}} w(R)x(R) \leq \ell \cdot \sum_{R' \in \mathcal{R}'} w'(R')x'(R'),$$

and that this inequality is tight in the case that no two descending paths of the same  $\ell$ -path appear in  $\mathcal{D}$ . To complete the proof of the lemma, we show that

$$\sum_{R \in \mathcal{R}} w(R)x(R) \leq \ell^2 \cdot \sum_{R \in \mathcal{R}} w(R)x^*(R).$$

For this, we define a fractional solution  $\bar{x}$  for the new instance  $(\mathcal{R}', \mathcal{B}', w')$  by setting  $\bar{x}(P_R^i) = \ell \cdot x^*(R)$  for each  $P_R^i \in \mathcal{R}'$ . To see that  $\bar{x}$  is feasible consider any blue  $\ell$ -path  $B \in \mathcal{B}$  and its representative  $P_B$ . By our selection of  $P_B$ , we have  $\sum_{R \in \mathcal{R}(P_B)} x^*(R) \geq 1/\ell$ , since otherwise  $x^*$  would not be feasible. It follows that  $\sum_{R' \in \mathcal{R}', R' \cap P_B} \bar{x}(R') \geq 1$  for each  $B' \in \mathcal{B}'$ . By Lemma 4, we know that the integral solution  $x'$  is an optimal fractional solution for  $(\mathcal{R}', \mathcal{B}', w')$ . Hence, the weight of  $\bar{x}$  is at least as high as the weight of  $x'$ . It follows that

$$\sum_{R \in \mathcal{R}} w(R)x(R) \leq \ell \cdot \sum_{R' \in \mathcal{R}'} w'(R')x'(R') \leq \ell \cdot \sum_{R' \in \mathcal{R}'} w'(R')\bar{x}(R').$$

Furthermore,

$$\sum_{R' \in \mathcal{R}'} w'(R')\bar{x}(R') = \sum_{R \in \mathcal{R}} \sum_{P \in \mathcal{B}} \frac{w(R)}{\ell} \cdot \ell \cdot x^*(R) = \ell \cdot \sum_{R \in \mathcal{R}} w(R) \cdot x^*(R)$$

and we are done.  $\square$

**Corollary 2.** MINIMUM DOMINATING SET in  $t$ -subtree graphs can be approximated within a factor of  $\ell(\mathcal{S})^2$ , where  $\mathcal{S}$  is the representation of the input graph, in polynomial-time.

We remark that our algorithm works for the case where  $\mathcal{R}$  contains  $r$ -paths and  $\mathcal{B}$  contains  $b$ -paths, and in this case the approximation ratio is  $r \cdot b$ . In fact, the instance that is generated by our second reduction (see Section 3.1) can be described by  $r = 1$  and  $b = t$ , and in this case the approximation ratio of our algorithm comes close to the  $\ell(\mathcal{S})$  lower bound given in Corollary 1.

#### 4. Maximum independent set

We next consider MAXIMUM INDEPENDENT SET. We present a  $2t$ -approximation algorithm for MAXIMUM INDEPENDENT SET that is based on the fractional local-ratio  $2t$ -approximation algorithm for  $t$ -interval graphs from [3].

Let  $G = (V, E)$  be a  $t$ -subtree graph, and let  $(\mathcal{T}, \mathcal{S})$  be its  $t$ -subtree representation, with  $\rho$  the root of  $\mathcal{T}$ . We define the root of a subtree  $T$  of  $\mathcal{T}$ , denoted  $\rho(T)$ , to be the node in  $T$  that is closest to  $\rho$  in  $\mathcal{T}$ . (Note that  $\rho(T)$  can be  $\rho$  itself.) We let  $\text{root}(\mathcal{S})$  denote the set of roots of subtrees that belong to  $t$ -subtrees of  $\mathcal{S}$ . That is,  $\text{root}(\mathcal{S}) = \{\rho(T) : T \in \mathcal{S}, S \in \mathcal{S}\}$ .

**Lemma 6.** Let  $S$  and  $S'$  be two intersecting  $t$ -subtrees in  $\mathcal{S}$ . Then, there exists some vertex  $\alpha \in \text{root}(\mathcal{S})$  such that  $\alpha \in S$  and  $\alpha \in S'$ .

**Proof.** If  $S$  and  $S'$  intersect, they have a pair of intersecting subtrees  $T \in S$  and  $T' \in S'$ . For these two trees we have either  $\rho(T) \in T'$  or  $\rho(T') \in T$ .  $\square$

Recall that, for a vertex  $v$  of  $G$ ,  $S_v$  denotes the  $t$ -subtree in  $\mathcal{S}$  corresponding to  $v$ , and  $w(v)$  denotes the weight of  $v$ . Lemma 6 implies that it is sufficient to look for intersections at  $\text{root}(\mathcal{S})$ . This allows us to formulate MAXIMUM INDEPENDENT SET in  $t$ -subtree graphs as the following integer program:

$$\begin{aligned} \max \quad & \sum_v w(v) \cdot x(v) \\ \text{s.t.} \quad & \sum_{v: \alpha \in S_v} x(v) \leq 1 \quad \forall \alpha \in \text{root}(\mathcal{S}) \\ & x(v) \in \{0, 1\} \quad \forall v \in V(G). \end{aligned} \tag{IS}$$

As usual,  $x(v)$  above denotes a variable corresponding to a vertex  $v$  of  $G$  which is to be interpreted as  $x(v) = 1$  if and only if the vertex  $v$  is chosen to the independent set. The constraints guarantee that at most one  $t$ -subtree contains  $\alpha$ , for every  $\alpha \in \text{root}(\mathcal{S})$ . Due to Lemma 6, it follows that at most one  $t$ -subtree is selected in any subset of intersecting  $t$ -subtrees. The linear programming relaxation of (IS) is obtained by replacing the constraints  $x(v) \in \{0, 1\}$  with  $0 \leq x(v) \leq 1$ , for every vertex  $v$  of  $G$ . Note that the integer (and linear) programming formulation for MAXIMUM INDEPENDENT SET in  $t$ -interval graphs given in [2] is identical to the one above when  $\mathcal{T}$  is a path.

Given a feasible solution  $x$  to the LP-relaxation of (IS), let us call the sum  $\sum_{u \in N[v]} x(u)$  the *fractional neighborhood* of a vertex  $v$  with respect to  $x$  (recall that  $N[v]$  denotes the set of neighbors of  $v$  in  $G$  including  $v$  itself). The fractional local-ratio based algorithm for MAXIMUM INDEPENDENT SET in  $t$ -interval graphs given in [3] essentially works by repeatedly selecting a vertex with a minimal fractional neighborhood, and deciding whether this vertex is in the solution independent set according to the recursive solution for  $G - N[v]$ . They proved that if, for any feasible solution  $x$ , there is always some vertex  $v$  with fractional neighborhood at most  $r$ , then their algorithm computes an  $r$ -approximate independent set. In order to extend their algorithm to  $t$ -subtree graphs, we prove the following lemma which generalizes the corresponding lemma for  $t$ -interval graphs from [3]:

**Lemma 7.** Given any feasible solution  $x$  of the LP-relaxation of (IS), there exists a vertex  $v$  of  $G$  with fractional neighborhood at most  $2t$  with respect to  $x$ .

**Proof.** In order to prove this lemma, it is enough to show that

$$\sum_v \sum_{u \in N[v]} x(v) \cdot x(u) = \sum_v x(v) \sum_{u \in N[v]} x(u) \leq 2t \cdot \sum_v x(v).$$

If  $\{u, v\} \in E$  then  $u \in N[v]$  and  $v \in N[u]$ . Therefore, the term  $x(v) \cdot x(u)$  is counted twice in the sum on the left hand side for every pair of neighboring vertices  $v$  and  $u$ . Furthermore, if  $\{u, v\} \in E$  then either there exists  $T \in S_v$  such that  $\rho(T) \in S_u$  or there exists  $T \in S_u$  such that  $\rho(T) \in S_v$ . Thus,

$$\sum_v \sum_{u \in N[v]} x(v) \cdot x(u) \leq 2 \cdot \sum_v \sum_{T \in S_v} \sum_{\rho(T) \in S_u} x(v) \cdot x(u).$$

Since  $x$  is a feasible solution of  $P$ , for every  $v$  and  $T \in S_v$ , we get that

$$\sum_{\rho(T) \in S_u} x(v) \cdot x(u) = x(v) \cdot \sum_{\rho(T) \in S_u} x(u) \leq x(v).$$

Therefore,

$$\sum_v \sum_{u \in N[v]} x(v) \cdot x(u) \leq 2 \cdot \sum_v \sum_{T \in S_v} x(v) = 2t \cdot \sum_v x(v),$$

and we are done.  $\square$

**Corollary 3.** MAXIMUM INDEPENDENT SET in  $t$ -subtree graphs can be approximated within a factor of  $2t$  in polynomial-time.

## 5. Minimum coloring and vertex cover

In this section we present approximation algorithms for MINIMUM COLORING and MINIMUM VERTEX COVER. The former achieves an approximation factor of  $2t$ , and the latter achieves a factor of  $2 - 1/t$ . Both algorithms rely on the following structural lemma for  $t$ -subtree graphs, which extends the corresponding lemma for  $t$ -interval graphs from [3]:

**Lemma 8.** Any  $t$ -subtree graph  $G$  with maximum clique size  $k$  can be colored in polynomial-time using at most  $2t(k - 1)$  colors.

**Proof.** Let  $G$  be a  $t$ -subtree, and let  $\mathcal{S}$  denote its  $t$ -subtree representation. Denote by  $G^*$  the intersection graph of  $\bigcup \mathcal{S}$ , the set of subtrees that appear in  $\mathcal{S}$ . Clearly,  $|V(G^*)| \leq t \cdot |V(G)|$ , since each vertex in  $G$  corresponds to at most  $t$  vertices in  $G^*$ , and  $|E(G)| \leq |E(G^*)|$ , since each edge in  $G$  corresponds to at least one edge in  $G^*$ . Notice that  $G^*$  is chordal and has maximum clique size at most  $k$ . Thus,  $|E(G^*)| < (k - 1)|V(G^*)|$ , as can be seen by counting all forwards edges in a simplicial ordering of  $G$ . Therefore,

$$|E(G)| \leq |E(G^*)| < (k - 1)|V(G^*)| \leq t(k - 1)|V(G)|.$$

It follows that the average degree of  $G$  (and its degeneracy) is less than  $2t(k - 1)$ , and so the standard greedy algorithm can be used to color  $G$  with at most  $2t(k - 1)$  colors.  $\square$

Since the maximum clique size of a graph  $G$  is a lower bound on the number of colors used in any coloring of  $G$ , and in particular in an optimal one, we obtain:

**Corollary 4.** MINIMUM COLORING in  $t$ -subtree graphs can be approximated within a factor of  $2t$  in polynomial-time.

Let us next consider MINIMUM VERTEX COVER. Here we propose an algorithm which consists of two stages. The first stage involves removing triangles from our input graph  $G$  by applying a technique originally introduced in [1] in order to remove short odd cycles. Using this technique, we obtain in polynomial-time a triangle-free subgraph  $G'$  of  $G$  such that any  $r$ -approximate vertex cover of  $G'$  can be easily transformed into a  $\max\{r, 1.5\}$ -approximate vertex cover of  $G$ . The same triangle-cleaning phase is also performed in [6] to approximate MINIMUM VERTEX COVER in  $t$ -interval graphs.

The second stage consists of using the algorithm of Hochbaum [13] which gives a factor of  $(2 - 2/c)$  for MINIMUM VERTEX COVER in graphs that can be colored in polynomial-time with  $c$  colors. Since  $G'$  is triangle-free, we have that it can be colored by  $2t$  colors due to Lemma 8. Hence, we get an algorithm that computes in polynomial-time a  $(2 - 1/t)$ -approximate vertex cover for the triangle-free subgraph  $G'$  produced in the first stage. As mentioned above, this vertex cover can be transformed into a vertex cover of  $G$  which is  $(2 - 1/t)$ -approximate for  $t \geq 2$ .

**Corollary 5.** MINIMUM VERTEX COVER in  $t$ -subtree graphs can be approximated within a factor of  $(2 - 1/t)$  in polynomial-time.

## 6. Maximum clique

In this section we present a  $(t^2 - t + 1)/2$ -approximation algorithm for MAXIMUM CLIQUE in  $t$ -subtree graphs.

We begin with the notion of a transversal. Let  $\mathcal{S}$  be a family of  $t$ -subtrees. A *transversal* of a subset  $\mathcal{S}' \subseteq \mathcal{S}$  is a set of vertices  $\{\alpha_1, \dots, \alpha_\tau\} \subset V(\mathcal{T})$  such that for every  $S \in \mathcal{S}'$  there is at least one  $\alpha_i \in \{\alpha_1, \dots, \alpha_\tau\}$  with  $\alpha_i \in S$ . Note that due to Lemma 6, we can assume that any transversal is a subset of  $\text{root}(\mathcal{S}')$ . The *transversal number* of  $\mathcal{S}'$  is the minimum size of any transversal of  $\mathcal{S}'$ .

A pairwise intersecting subset of  $\mathcal{S}$  (i.e., a clique in the  $t$ -subtree graph) is called a  $\tau$ -clique if it has transversal number equal to  $\tau$ . (Note that a  $\tau$ -clique is not a clique with  $\tau$  vertices.). Berger [4] proved upper bounds on the transversal number of any pairwise intersecting family of  $t$ -subtrees.

**Lemma 9** ([4]). Let  $\mathcal{S}$  be a family of  $t$ -subtrees. Any pairwise intersecting subset  $\mathcal{S}' \subseteq \mathcal{S}$  is a  $\tau$ -clique for  $\tau \leq t^2 - t + 1$ .

As in  $t$ -interval families, the maximum weight 2-clique in  $\mathcal{S}$  can be computed in polynomial-time due to a couple of simple observations.

**Lemma 10.** A maximum weight 2-clique in  $\mathcal{S}$  can be computed in polynomial time.



**Proof.** Consider a pair of vertices  $\alpha, \beta \in \text{root}(\mathcal{S})$ , and let  $\mathcal{S}' = \{S \in \mathcal{S} : \alpha \in S \text{ or } \beta \in S\}$ . The intersection graph  $G_{\mathcal{S}'}$  is the complement of a bipartite graph, since both  $\{S \in \mathcal{S} : \alpha \in S\}$  and  $\{S \in \mathcal{S} : \beta \in S\}$  are pairwise intersecting. Since MAXIMUM INDEPENDENT SET is polynomial-time solvable in bipartite graphs, we can compute the maximum weight clique in  $G_{\mathcal{S}'}$  in polynomial-time. Thus, by iterating over all  $O(n^2)$  pairs of vertices in  $\text{root}(\mathcal{S})$ , we can compute a maximum weight 2-clique in  $\mathcal{S}$  in polynomial time.  $\square$

Combining both lemmas above, we obtain:

**Corollary 6.** MAXIMUM CLIQUE in  $t$ -subtree graphs can be approximated in polynomial-time within a factor of  $(t^2 - t + 1)/2$ .

**Proof.** Let  $\mathcal{C}$  be a maximum pairwise intersecting subset. Due to Lemma 9 we know that the transversal number of  $\mathcal{C}$  is at most  $t^2 - t + 1$ . It follows that there exist two vertices that are contained in at least  $2|\mathcal{C}|/(t^2 - t + 1)$  subtrees from  $\mathcal{C}$ . Hence, a maximum weight 2-clique in  $\mathcal{S}$  is  $(t^2 - t + 1)/2$ -approximate.  $\square$

## Acknowledgements

This paper is dedicated to Martin C. Golumbic on his 60th birthday. Both authors are grateful for having the privilege of working at the Caesarea Edmond Benjamin de Rothschild Foundation Institute at the University of Haifa, headed and founded by Martin. We are grateful not only for the many seminars and research-related discussions we had with Martin along the years (in particular, the first author was introduced to the topic of geometric intersection graphs by Martin), but also for the numerous hallway discussions with him about life and academia. Martin has definitely been a cornerstone in both of our academic careers.

The first author was supported by the Adams Fellowship of the Israel Academy of Sciences and Humanities.

## References

- [1] R. Bar-Yehuda, S. Even, A local-ratio theorem for approximating the weighted vertex cover problem, *Annals of Discrete Mathematics* 25 (1985) 27–46.
- [2] R. Bar-Yehuda, M.M. Halldórsson, J. Naor, H. Shachnai, I. Shapira, Scheduling split intervals, in: 13th Annual ACM–SIAM Symposium on Discrete Algorithms, 2002, pp. 732–741.
- [3] R. Bar-Yehuda, M.M. Halldórsson, J. Naor, H. Shachnai, I. Shapira, Scheduling split intervals, *SIAM Journal on Computing* 36 (1) (2006) 1–15.
- [4] E. Berger, KKM—a topological approach for trees, *Combinatorica* 25 (1) (2004) 1–18.
- [5] K.S. Booth, J.H. Johnson, Dominating set in chordal graphs, *SIAM Journal on Computing* 11 (1) (1982) 191–199.
- [6] A. Butman, D. Hermelin, M. Lewenstein, D. Rawitz, Optimization problems in multiple-interval graphs, in: 18th Annual ACM–SIAM Symposium on Discrete Algorithms, 2007, pp. 268–277.
- [7] I. Dinur, V. Guruswami, S. Khot, O. Regev, A new multilayered PCP and the hardness of hypergraph vertex cover, *SIAM Journal on Computing* 34 (5) (2005) 1129–1146.
- [8] M.R. Garey, D.S. Johnson, L. Stockmeyer, Some simplified NP-complete graph problems, *Theoretical Computer Science* 1 (1976) 237–267.
- [9] F. Gavril, Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph, *SIAM Journal on Computing* 1 (2) (1972) 180–187.
- [10] F. Gavril, The intersection graphs of subtrees in trees are exactly the chordal graphs, *Journal of Combinatorial Theory, Series B* 16 (1) (1974) 47–56.
- [11] M.C. Golumbic, *Algorithmic graph theory and perfect graphs*, Academic Press, New York, 1980.
- [12] J.R. Griggs, D.B. West, Extremal values of the interval number of a graph, *SIAM Journal on Algebraic and Discrete Methods* 1 (1) (1980) 1–14.
- [13] D.S. Hochbaum, Efficient bounds for the stable set, vertex cover and set packing problems, *Discrete Applied Mathematics* 6 (1983) 243–254.
- [14] C. Papadimitriou, M. Yannakakis, Optimization, approximation, and complexity classes, *Journal of Computer and Systems Sciences* 43 (1991) 425–440.
- [15] O. Parekh, D. Segev, Path hitting in acyclic graphs, *Algorithmica* 52 (4) (2008) 466–486.
- [16] D.B. West, D.B. Shmoys, Recognizing graphs with fixed interval number is NP-complete, *Discrete Applied Mathematics* 8 (1984) 295–305.